



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/839,644	04/20/2001	Dietrich Charisius	30013630-0019	2137

23485 7590 05/13/2004

JINAN GLASGOW
P O BOX 28539
RALEIGH, NC 276118539

EXAMINER

INGBERG, TODD D

ART UNIT PAPER NUMBER

2124

DATE MAILED: 05/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/839,644

Applicant(s)

CHARISIUS ET AL.

Examiner

Todd Ingberg

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 22 March 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-225 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-225 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
- a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 4,6.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other:

Art Unit: 2124

DETAILED ACTION

Claims 1 – 225 have been examined.

Information Disclosure Statement

1. The Information Disclosure Statement filed July 16, 2001 and August 13, 2001 have been considered.

Priority

2. Claim to priority is still being investigated.

Specification

3. Examiner is requesting the Applicant complete the Cross Reference to Related Applications section on pages 1 and 2 of the Specification.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1- 225 are rejected under 35 U.S.C. 102(b) as being anticipated by Template Software's commercial product line.

The **Template** product line contains:

The SNAP programming language (Not used in this Office Action)

The Workflow Template (Two manuals used)

The Web Component (Not used in this Office Action)

These three layered products work together.

Art Unit: 2124

The documentation sets for the products contains the following manuals.

SNAP released June 1997

SNAP Language Reference (Not used in this Office Action)

Using the SNAP Language (Not used in this Office Action)

Using the SNAP Communication Component (Not used in this Office Action)

Using the SNAP Graphic User Interface Component (Not used in this Office Action)

Getting Started with SNAP (Not used in this Office Action)

Using the SNAP Display Editors (Not used in this Office Action)

SNAP Class Library Reference (Not used in this Office Action)

Using the SNAP External Application Software Component (Not used in this Office Action)

Using the SNAP Development Environment (Referred to as **SNAP**)

SNAP Module Library Reference (Not used in this Office Action)

Using the SNAP Permanent Storage Component (Not used in this Office Action)

Workflow released September 1997

Developing a WFT Workflow System (Not used in this Office Action)

Using the WFT Development Environment (Not used in this Office Action)

WFT Library Reference (Not used in this Office Action)

Web Component

Using the Web Component (Not used in this Office Action)

Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press

Art Unit: 2124

release announcing version 8.0 these considered prior art under *In re Epstein* 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date.

SNAP anticipates an object oriented programming environment (OO-CASE tool) with an object model editor (**SNAP**, page 3-6). With in the Object model editor the user is able to link classes. The links include inheritance and relation lines (**SNAP**, page 3-9). **SNAP** support a plurality of dependencies (**SNAP**, page 3-14). In addition to the lines representing a form of interface the functions also interface (page 3-44 parameters). One of ordinary skill in the art should know messaging is inherent in object technology is also a form of interface (**SNAP**, page 3-44 functions are methods also can be called packages which support method calls). Furthermore, the **SNAP** environment provides for error messages when rules in the object model editor have been violated (**SNAP**, pages 2-15 and 2-17). **SNAP** also supports a class (node) representing multiple links (**SNAP**, page 3-15 down arrow indicating child classes are present but not shown). The ability to declare variables (attributes) is preformed by the Attribute tab (**SNAP**, page 3-40). The code between two classes linked through inheritance is performed dynamically and is visible the class editor (**SNAP**, page 3-31 – Inherits From). It should be noted that the same interpreted environment generates the error messages above. Also the inherited attributes and class attributes are visible on **SNAP** page 3-40. Local attributes to functions/packages/methods are shown on **SNAP** page 3-44

Conclusion

3. ***Undue multiplicity*** is an unpopular rejection within the Patent Office at this time. However, the Examiner did not see the value in 225 claims. The claims are repeated below.

Claim 1

Art Unit: 2124

A method in a data processing system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the method comprising the steps of: displaying a graphical representation of the dependent node; displaying a graphical representation of the defining node; receiving an indication to identify a dependency between the dependent node and the defining node; and in response to receiving the indication to identify the dependency, representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node.

Claim 2

The method of claim 1, wherein the plurality of dependencies is represented as a single link.

Claim 3

The method of claim 1, wherein each node comprises an element.

Claim 4

The method of claim 3, wherein one of the plurality of dependencies comprises a use of the defining node element by the dependent node element.

Claim 5

The method of claim 3, wherein one of the plurality of dependencies comprises a declaration of the defining node element by the dependent node element.

Claim 6

The method of claim 3, wherein one of the plurality of dependencies comprises a call to a method of the defining node element by the dependent node element.

Claim 7

The method of claim 3, wherein one of the plurality of dependencies comprises a local variable definition using the defining node element in a method of the dependent node element.

Claim 8

The method of claim 3, wherein the dependent node element comprises a class.

Claim 9

The method of claim 3, wherein the dependent node element comprises an interface.

Claim 10

The method of claim 3, wherein the defining node element comprises a class.

Claim 11

The method of claim 3, wherein the defining node element comprises an interface.

Claim 12

The method of claim 1, wherein the dependent node comprises a class.

Claim 13

The method of claim 1, wherein the dependent node comprises an interface.

Claim 14

The method of claim 1, wherein the dependent node comprises a package.

Claim 15

The method of claim 14, wherein the package comprises a plurality of elements.

Claim 16

The method of claim 15, wherein one of the plurality of elements comprises a class.

Claim 17

The method of claim 15, wherein one of the plurality of elements comprises an interface.

Claim 18

The method of claim 1, wherein the defining node comprises a class.

Claim 19

The method of claim 1, wherein the defining node comprises an interface.

Claim 20

The method of claim 1, wherein the defining node comprises a package.

Claim 21

The method of claim 20, wherein the package comprises a plurality of elements.

Claim 22

The method of claim 21, wherein one of the plurality of elements comprises a class.

Claim 23

The method of claim 21, wherein one of the plurality of elements comprises an interface.

Claim 24

A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: displaying a graphical representation of the plurality of nodes; determining whether the code corresponding to a first of the plurality of nodes contains a first use of a second of the plurality of nodes; and when it is determined that the code corresponding to the first node contains the first use of the second node, determining whether the code corresponding to the first node contains a second use of the second node; and when it is determined that the code corresponding to the first node contains the second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Art Unit: 2124

Claim 25

The method of claim 24, wherein the first use comprises a declaration.

Claim 26

The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an attribute declaration that uses the second node.

Claim 27

The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an initializer of an attribute declaration that uses the second node.

Claim 28

The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an argument parameter of a method that uses the second node.

Claim 29

The method of claim 24, wherein the first use comprises a method call.

Claim 30

The method of claim 24, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

Claim 31

The method of claim 24, wherein the first node comprises a class.

Claim 32

The method of claim 24, wherein the first node comprises an interface.

Claim 33

The method of claim 24, wherein the first node comprises a package.

Claim 34

The method of claim 24, wherein the second node comprises a class.

Claim 35

The method of claim 24, wherein the second node comprises an interface.

Claim 36

The method of claim 24, wherein the second node comprises a package.

Art Unit: 2124

Claim 37

A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of displaying a graphical representation of the plurality of nodes; determining whether the code corresponding to a first of the plurality of nodes contains a declaration of the second node; and when it is determined that the code corresponding to the first node contains the declaration of the second node, determining whether the code corresponding to the first node contains another declaration of the second node; and when it is determined that the code corresponding to the first node contains the other declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 38

The method of claim 37, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, determining whether the code corresponding to the first node contains a call to a method of the second node; and when it is determined that the code corresponding to the first node contains the call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 39

The method of claim 37, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 40

The method of claim 37, wherein the first node comprises a class.

Claim 41

The method of claim 37, wherein the first node comprises an interface.

Claim 42

The method of claim 37, wherein the first node comprises a package.

Claim 43

The method of claim 37, wherein the second node comprises a class.

Claim 44

The method of claim 37, wherein the second node comprises an interface.

Art Unit: 2124

Claim 45

The method of claim 37, wherein the second node comprises a package.

Claim 46

A method in a data processing system having a first node and a second node, each of the nodes having corresponding code, the second node having a method, the method comprising the steps of: displaying a graphical representation of the first node and the second node; determining whether the code corresponding to the first node contains a call to the method of the second node; and when it is determined that the code corresponding to the first node contains the call to the method of the second node, determining whether the code corresponding of the first node contains another call to the method of the second node; and when it is determined that the code corresponding to the first node contains the other call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 47

The method of claim 46, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 48

The method of claim 46, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, determining whether the code corresponding to the first node contains a declaration of the second node; and when it is determined that the code corresponding to the first node contains the declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 49

The method of claim 46, wherein the first node comprises a class.

Claim 50

The method of claim 46, wherein the first node comprises an interface.

Claim 51

The method of claim 46, wherein the first node comprises a package.

Claim 52

The method of claim 46, wherein the second node comprises a class.

Art Unit: 2124

Claim 53

The method of claim 46, wherein the second node comprises an interface.

Claim 54

The method of claim 46, wherein the second node comprises a package.

Claim 55

A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes; determining whether the code corresponding to the first node contains a first use and a second use of the second node; and when it is determined that the code corresponding to the first node contains the first use and the second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 56

The method of claim 55, wherein the first use comprises a declaration.

Claim 57

The method of claim 55, wherein the first use comprises a method call.

Claim 58

The method of claim 55, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

Claim 59

The method of claim 55, wherein the first node comprises a class.

Claim 60

The method of claim 55, wherein the first node comprises an interface.

Claim 61

The method of claim 55, wherein the first node comprises a package.

Claim 62

The method of claim 55, wherein the second node comprises a class.

Claim 63

The method of claim 55, wherein the second node comprises an interface.

Claim 64

The method of claim 55, wherein the second node comprises a package.

Art Unit: 2124

Claim 65

A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes; determining whether the code corresponding to the first node contains a declaration of the second node; and when it is determined that the code corresponding to the first node contains the declaration of the second node, determining whether the code corresponding to the first node contains another declaration of the second node; and when it is determined that the code corresponding to the first node contains the other declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 66

The method of claim 65, wherein the first node comprises a class.

Claim 67

The method of claim 65, wherein the first node comprises an interface.

Claim 68

The method of claim 65, wherein the first node comprises a package.

Claim 69

The method of claim 65, wherein the second node comprises a class.

Claim 70

The method of claim 65, wherein the second node comprises an interface.

Claim 71

The method of claim 65, wherein the second node comprises a package.

Claim 72

A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes, wherein the second node has a method; determining whether the code corresponding to the first node contains a call to the method of the second node; and when it is determined that the code corresponding to the first node contains the call to the method of the second node, determining whether the code corresponding of the first node contains another call to the method of the second node; and when it is determined that the code corresponding to the first node contains another call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 73

The method of claim 72, wherein the first node comprises a class.

Art Unit: 2124

Claim 74

The method of claim 72, wherein the first node comprises an interface.

Claim 75

The method of claim 72, wherein the first node comprises a package.

Claim 76

The method of claim 72, wherein the second node comprises a class.

Claim 77

The method of claim 72, wherein the second node comprises an interface.

Claim 78

The method of claim 72, wherein the second node comprises a package.

Claim 79

A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes; determining whether the code corresponding to the first node contains a use of the second node; and when it is determined that the code corresponding to the first node contains the use of the second node, displaying the usage of the second node by the first node.

Claim 80

The method of claim 79, wherein each node comprises an element.

Claim 81

The method of claim 79, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the first node element.

Claim 82

The method of claim 81, further comprising the step of displaying the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

Claim 83

The method of claim 79, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the second node element.

Claim 84

Art Unit: 2124

The method of claim 83, further comprising the step of displaying the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

Claim 85

The method of claim 79, wherein the first node comprises a class.

Claim 86

The method of claim 79, wherein the first node comprises an interface.

Claim 87

The method of claim 79, wherein the first node comprises a package.

Claim 88

The method of claim 79, wherein the second node comprises a class.

Claim 89

The method of claim 79, wherein the second node comprises an interface.

Claim 90

The method of claim 79, wherein the second node comprises a package.

Claim 91

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the method comprising the steps of: displaying a graphical representation of the dependent node; displaying a graphical representation of the defining node; receiving an indication to identify a dependency between the dependent node and the defining node; and in response to receiving the indication to identify the dependency, representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node.

Claim 92

The computer-readable medium of claim 91, wherein the plurality of dependencies is represented as a single link.

Claim 93

The computer-readable medium of claim 91, wherein each node comprises an element.

Claim 94

The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a use of the defining node element by the dependent node element.

Claim 95

Art Unit: 2124

The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a declaration of the defining node element by the dependent node element.

Claim 96

The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a call to a method of the defining node element by the dependent node element.

Claim 97

The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a local variable definition using the defining node element in a method of the dependent node element.

Claim 98

The computer-readable medium of claim 93, wherein the dependent node element comprises a class.

Claim 99

The computer-readable medium of claim 93, wherein the dependent node element comprises an interface.

Claim 100

The computer-readable medium of claim 93, wherein the defining node element comprises a class.

Claim 101

The computer-readable medium of claim 93, wherein the defining node element comprises an interface.

Claim 102

The computer-readable medium of claim 91, wherein the dependent node comprises a class.

Claim 103

The computer-readable medium of claim 91, wherein the dependent node comprises an interface.

Claim 104

The computer-readable medium of claim 91, wherein the dependent node comprises a package.

Claim 105

The computer-readable medium of claim 104, wherein the package comprises a plurality of elements.

Claim 106

The computer-readable medium of claim 105, wherein one of the plurality of elements comprises a class.

Art Unit: 2124

Claim 107

The computer-readable medium of claim 105, wherein one of the plurality of elements comprises an interface.

Claim 108

The computer-readable medium of claim 91, wherein the defining node comprises a class.

Claim 109

The computer-readable medium of claim 91, wherein the defining node comprises an interface.

Claim 110

The computer-readable medium of claim 91, wherein the defining node comprises a package.

Claim 111

The computer-readable medium of claim 110, wherein the package comprises a plurality of elements.

Claim 112

The computer-readable medium of claim 111, wherein one of the plurality of elements comprises a class.

Claim 113

The computer-readable medium of claim 111, wherein one of the plurality of elements comprises an interface.

Claim 114

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: displaying a graphical representation of the plurality of nodes; determining whether the code corresponding to a first of the plurality of nodes contains a first use of a second of the plurality of nodes; and when it is determined that the code corresponding to the first node contains the first use of the second node, determining whether the code corresponding to the first node contains a second use of the second node; and when it is determined that the code corresponding to the first node contains the second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 115

The computer-readable medium of claim 114, wherein the first use comprises a declaration.

Claim 116

The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of

Art Unit: 2124

searching the code corresponding to the first node for an attribute declaration that uses the second node.

Claim 117

The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an initializer of an attribute declaration that uses the second node.

Claim 118

The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an argument parameter of a method that uses the second node.

Claim 119

The computer-readable medium of claim 114, wherein the first use comprises a method call.

Claim 120

The computer-readable medium of claim 114, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

Claim 121

The computer-readable medium of claim 114, wherein the first node comprises a class.

Claim 122

The computer-readable medium of claim 114, wherein the first node comprises an interface.

Claim 123

The computer-readable medium of claim 114, wherein the first node comprises a package.

Claim 124

The computer-readable medium of claim 114, wherein the second node comprises a class.

Claim 125

The computer-readable medium of claim 114, wherein the second node comprises an interface.

Claim 126

The computer-readable medium of claim 114, wherein the second node comprises a package.

Claim 127

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: displaying a graphical

Art Unit: 2124

representation of the plurality of nodes; determining whether the code corresponding to a first of the plurality of nodes contains a declaration of the second node; and when it is determined that the code corresponding to the first node contains the declaration of the second node, determining whether the code corresponding to the first node contains another declaration of the second node; and when it is determined that the code corresponding to the first node contains the other declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 128

The computer-readable medium of claim 127, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the method further comprises the steps of determining whether the code corresponding to the first node contains a call to a method of the second node; and when it is determined that the code corresponding to the first node contains the call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 129

The computer-readable medium of claim 127, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the method further comprises the steps of: determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 130

The computer-readable medium of claim 127, wherein the first node comprises a class.

Claim 131

The computer-readable medium of claim 127, wherein the first node comprises an interface.

Claim 132

The computer-readable medium of claim 127, wherein the first node comprises a package.

Claim 133

The computer-readable medium of claim 127, wherein the second node comprises a class.

Claim 134

The computer-readable medium of claim 127, wherein the second node comprises an interface.

Claim 135

The computer-readable medium of claim 127, wherein the second node comprises a package.

Art Unit: 2124

Claim 136

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a first node and a second node, each of the nodes having corresponding code, the second node having a method, the method comprising the steps of displaying a graphical representation of the first node and the second node; determining whether the code corresponding to the first node contains a call to the method of the second node; and when it is determined that the code corresponding to the first node contains the call to the method of the second node, determining whether the code corresponding of the first node contains another call to the method of the second node; and when it is determined that the code corresponding to the first node contains the other call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 137

The computer-readable medium of claim 136, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the method further comprises the steps of determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 138

The computer-readable medium of claim 136, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the method further comprises the steps of determining whether the code corresponding to the first node contains a declaration of the second node; and when it is determined that the code corresponding to the first node contains the declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 139

The computer-readable medium of claim 136, wherein the first node comprises a class.

Claim 140

The computer-readable medium of claim 136, wherein the first node comprises an interface.

Claim 141

The computer-readable medium of claim 136, wherein the first node comprises a package.

Claim 142

The computer-readable medium of claim 136, wherein the second node comprises a class.

Claim 143

Art Unit: 2124

The computer-readable medium of claim 136, wherein the second node comprises an interface.

Claim 144

The computer-readable medium of claim 136, wherein the second node comprises a package.

Claim 145

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes; determining whether the code corresponding to the first node contains a first use and a second use of the second node; and when it is determined that the code corresponding to the first node contains the first use and the second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 146

The computer-readable medium of claim 145, wherein the first use comprises a declaration.

Claim 147

The computer-readable medium of claim 145, wherein the first use comprises a method call.

Claim 148

The computer-readable medium of claim 145, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

Claim 149

The computer-readable medium of claim 145, wherein the first node comprises a class.

Claim 150

The computer-readable medium of claim 145, wherein the first node comprises an interface.

Claim 151

The computer-readable medium of claim 145, wherein the first node comprises a package.

Claim 152

The computer-readable medium of claim 145, wherein the second node comprises a class.

Claim 153

The computer-readable medium of claim 145, wherein the second node comprises an interface.

Claim 154

The computer-readable medium of claim 145, wherein the second node

Art Unit: 2124

Claim 155

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes; determining whether the code corresponding to the first node contains a declaration of the second node; and when it is determined that the code corresponding to the first node contains the declaration of the second node, determining whether the code corresponding to the first node contains another declaration of the second node; and when it is determined that the code corresponding to the first node contains the other declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 156

The computer-readable medium of claim 155, wherein the first node comprises a class.

Claim 157

The computer-readable medium of claim 155, wherein the first node comprises an interface.

Claim 158

The computer-readable medium of claim 155, wherein the first node comprises a package.

Claim 159

The computer-readable medium of claim 155, wherein the second node comprises a class.

Claim 160

The computer-readable medium of claim 155, wherein the second node comprises an interface.

Claim 161

The computer-readable medium of claim 155, wherein the second node comprises a package.

Claim 162

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of: receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes, wherein the second node has a method; determining whether the code corresponding to the first node contains a call to the method of the second node; and when it is determined that the code corresponding to the first node contains the call to the method of the second node, determining whether the code corresponding of the first node contains another call to the method of the second node; and when it is determined that the code corresponding to the first node contains another call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Art Unit: 2124

Claim 163

The computer-readable medium of claim 162, wherein the first node comprises a class.

Claim 164

The computer-readable medium of claim 162, wherein the first node comprises an interface.

Claim 165

The computer-readable medium of claim 162, wherein the first node comprises a package.

Claim 166

The computer-readable medium of claim 162, wherein the second node comprises a class.

Claim 167

The computer-readable medium of claim 162, wherein the second node comprises an interface.

Claim 168

The computer-readable medium of claim 162, wherein the second node comprises a package.

Claim 169

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of receiving an indication of a first of the plurality of nodes; receiving an indication of a second of the plurality of nodes; determining whether the code corresponding to the first node contains a use of the second node; and when it is determined that the code corresponding to the first node contains the use of the second node, displaying the usage of the second node by the first node.

Claim 170

The computer-readable medium of claim 169, wherein each node comprises an element.

Claim 171

The computer-readable medium of claim 169, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the first node element.

Claim 172

The computer-readable medium of claim 171, wherein the method further comprises the step of displaying the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

Claim 173

The computer-readable medium of claim 169, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the second node element.

Art Unit: 2124

Claim 174

The computer-readable medium of claim 173, wherein the method further comprises the step of displaying the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

Claim 175

The computer-readable medium of claim 169, wherein the first node comprises a class.

Claim 176

The computer-readable medium of claim 169, wherein the first node comprises an interface.

Claim 177

The computer-readable medium of claim 169, wherein the first node comprises a package.

Claim 178

The computer-readable medium of claim 169, wherein the second node comprises a class.

Claim 179

The computer-readable medium of claim 169, wherein the second node comprises an interface.

Claim 180

The computer-readable medium of claim 169, wherein the second node comprises a package.

Claim 181

A data processing system comprising: a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code; a memory device further comprising a program that displays a graphical representation of the plurality of nodes, that determines whether the code corresponding to a first of the plurality of nodes contains a declaration of the second node, and when it is determined that the code corresponding to the first node contains the declaration of the second node, the program determines whether the code corresponding to the first node contains another declaration of the second node, and when it is determined that the code corresponding to the first node contains the other declaration of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and a processor for running the program.

Claim 182

The data processing system of claim 181, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the program determines whether the code corresponding to the first node contains a call to a method of the second node, and when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program displays a dependency link

Art Unit: 2124

between the graphical representation of the first node and the graphical representation of the second node.

Claim 183

The data processing system of claim 181, wherein when it is determined that the code corresponding to the first node does not contain the other declaration of the second node, the program determines whether the code corresponding to the first node comprises a method having a local variable definition using the second node, and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 184

The data processing system of claim 181, wherein the first node comprises a class.

Claim 185

The data processing system of claim 181, wherein the first node comprises an interface.

Claim 186

The data processing system of claim 181, wherein the first node comprises a package.

Claim 187

The data processing system of claim 181, wherein the second node comprises a class.

Claim 188

The data processing system of claim 181, wherein the second node comprises an interface.

Claim 189

The data processing system of claim 181, wherein the second node comprises a package.

Claim 190

A data processing system comprising: a secondary storage device further comprising a first node and a second node, each of the nodes having corresponding code, the second node having a method; a memory device further comprising a program that displays a graphical representation of the first node and the second node, that determines whether the code corresponding to the first node contains a call to the method of the second node, and when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program determines whether the code corresponding of the first node contains another call to the method of the second node, and when it is determined that the code corresponding to the first node contains the other call to the method of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and a processor for running the program.

Claim 191

Art Unit: 2124

The data processing system of claim 190, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the program determines whether the code corresponding to the first node comprises a method having a local variable definition using the second node, and when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 192

The data processing system of claim 190, wherein when it is determined that the code corresponding to the first node does not contain the other call to the method of the second node, the program determines whether the code corresponding to the first node contains a declaration of the second node, and when it is determined that the code corresponding to the first node contains the declaration of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

Claim 193

The data processing system of claim 190, wherein the first node comprises a class.

Claim 194

The data processing system of claim 190, wherein the first node comprises an interface.

Claim 195

The data processing system of claim 190, wherein the first node comprises a package.

Claim 196

The data processing system of claim 190, wherein the second node comprises a class.

Claim 197

The data processing system of claim 190, wherein the second node comprises an interface.

Claim 198

The data processing system of claim 190, wherein the second node comprises a package.

Claim 199

A data processing system comprising: a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code; a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that receives an indication of a second of the plurality of nodes, that determines whether the code corresponding to the first node contains a declaration of the second node, and when it is determined that the code corresponding to the first node contains the declaration of the second node, the program determines whether the code corresponding to the first node contains another declaration of the second node, and when it is determined that the code corresponding to the first node contains the other declaration of the second node, the program displays a dependency link

Art Unit: 2124

between the graphical representation of the first node and the graphical representation of the second node; and a processor for running the program.

Claim 200

The data processing system of claim 199, wherein the first node comprises a class.

Claim 201

The data processing system of claim 199, wherein the first node comprises an interface.

Claim 202

The data processing system of claim 199, wherein the first node comprises a package.

Claim 203

The data processing system of claim 199, wherein the second node comprises a class.

Claim 204

The data processing system of claim 199, wherein the second node comprises an interface.

Claim 205

The data processing system of claim 199, wherein the second node comprises a package.

Claim 206

A data processing system comprising: a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code; a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that receives an indication of a second of the plurality of nodes, wherein the second node has a method, that determines whether the code corresponding to the first node contains a call to the method of the second node, and when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program determines whether the code corresponding of the first node contains another call to the method of the second node, and when it is determined that the code corresponding to the first node contains another call to the method of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and a processor for running the program.

Claim 207

The data processing system of claim 206, wherein the first node comprises a class.

Claim 208

The data processing system of claim 206, wherein the first node comprises an interface.

Claim 209

The data processing system of claim 206, wherein the first node comprises a package.

Art Unit: 2124

Claim 210

The data processing system of claim 206, wherein the second node comprises a class.

Claim 211

The data processing system of claim 206, wherein the second node comprises an interface.

Claim 212

The data processing system of claim 206, wherein the second node comprises a package.

Claim 213

A data processing system comprising: a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code; a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that receives an indication of a second of the plurality of nodes, that determines whether the code corresponding to the first node contains a use of the second node, and when it is determined that the code corresponding to the first node contains the use of the second node, the program displays the usage of the second node by the first node; and a processor for running the program.

Claim 214

The data processing system of claim 213, wherein each node comprises an element.

Claim 215

The data processing system of claim 213, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the program further displays the first node element.

Claim 216

The data processing system of claim 215, wherein the program further displays the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

Claim 217

The data processing system of claim 213, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the program further displays the second node element.

Claim 218

The data processing system of claim 217, wherein the program further displays the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

Claim 219

The data processing system of claim 213, wherein the first node comprises a class.

Art Unit: 2124

Claim 220

The data processing system of claim 213, wherein the first node comprises an interface.

Claim 221

The data processing system of claim 213, wherein the first node comprises a package.

Claim 222

The data processing system of claim 213, wherein the second node comprises a class.

Claim 223

The data processing system of claim 213, wherein the second node comprises an interface.

Claim 224

The data processing system of claim 213, wherein the second node comprises a package.

Claim 225

A system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the system comprising: means for displaying a graphical representation of the dependent node; means for displaying a graphical representation of the defining node; means for receiving an indication to identify a dependency between the dependent node and the defining node; and means for representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node in response to receiving the indication to identify the dependency.

4. In the broadest reasonable interpretation in view of the Specification the terms class/package and node are taught in the prior art. The Specification mentions UML but the claim language is silent.

Correspondence Information

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Todd Ingberg** whose telephone number is (703) 305-9775. The examiner can normally be reached during the following hours:

Monday	Tuesday	Wednesday	Thursday	Friday
6:15 – 1:30	6:15- 3:45	6:15 – 4:45	6:15-3:45	6:15-130

This schedule began December 1, 2003 and is subject to change.

Art Unit: 2124

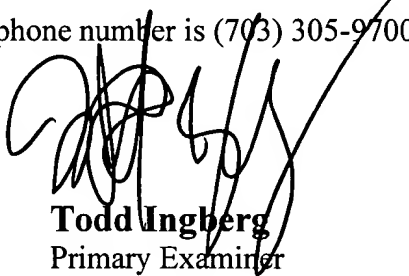
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Kakali Chaki** can be reached on (703) 305-9662. Please, note that as of August 4, 2003 the **FAX number** changed for the organization where this application or proceeding is assigned is **(703) 872-9306**.

Also, be advised the United States Patent Office **new address** is

Post Office Box 1450

Alexandria, Virginia 22313-1450

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9700.



Todd Ingberg
Primary Examiner
Art Unit 2124
May 10, 2004